Sinonym-Generator

Membri del gruppo: Lovreglio Giuseppe 708245
Link Repository: https://github.com/BigPyno/synonym-generator.git

Il progetto consiste in un' applicazione che, data una frase in input, è in grado di generare sinonimi per le parole che la formano. Per un corretto utilizzo dell'applicazione è necessario inserire frasi in lingua inglese.

Funzionamento
L'utente inserisce una frase in lingua inglese e clicca il tasto **Invia**.
Il sistema prende in input la frase ed esegue la correzione sintattica di tutte le parole che la compongono. Dopodichè il sistema consiglia 5 sinonimi per ogni parola che la compone (fatta eccezione per stopwords e punteggiatura).
L'utente può cliccare sul sinonimo che vuole utilizzare per inserirlo all'interno della frase al posto della parola corrispondente.
Nel caso la frase contenga parole che il sistema non conosce, il sistema non effettua alcuna predizione per queste ultime ma,all'utilizzo successivo, il sistema chiederà all'utente se desidera aggiornare la conoscenza del sistema inserendo le nuove parole nel modello. In caso di risposta affermativa, partirà il training con le nuove parole.
E' inoltre possibile visualizzare graficamente il modello di clustering cliccando sul tasto **Cluster**

Scelte di progettazione
Per lo sviluppo dell'applicazione è stato scelto il linguaggio Python.
Come sorgente di conoscenza sono stati ragruppati e poi utilizzati due dataset diversi:

- un file .csv contenente numerose recensioni di film scaricato dal sito IMDB.
- *Cornell Movie-Dialogs Corpus*, ovvero un insieme di conversazioni estratti da diversi film.

Innanzitutto sono state effetuate operazioni di text processing come conversione in minuscolo,tokenizzazione, rimozione della punteggiatura e rimozione delle stopwords.
Al termine di questo processo, le frasi tokenizzate sono state utilizzate come input per creare il modello di apprendimento. E' stato utilizzato un approccio di apprendimento non supervisionato, nello specifico soft-clustering.
Per fare ciò ad ogni parola è stata applicata una trasformazione Word2Vec, che appunto trasforma ogni parola in un vettore di n dimensioni (nel nostro caso 100 dimensioni). Dopo aver addestrato il modello, il sistema è in grado di calcolare la similarità tra due parole in base alla loro distanza nel modello.
Il modello,formato da 115172 parole, viene serializzato e salvato su file.

Infine è stata creata una interfaccia grafica con cui è possibile utilizzare l'applicazione.

La correzione sintattica delle parole avviene controllando la distanza sintattica che le stesse hanno da una lista 466000 parole della lingua inglese, contenute nel file *word.txt*
Le nuove parole inserita vengono salvato sul file *nuoveparole.txt*, il cui contenuto viene controllato ad ogni avvio.
Il grafico del modello viene creato nel momento dell'addestramento e viene salvato sul file *grafico.html*
Il tempo impiegato per l'addestramento sulla macchina di test è di circa 160 secondi, mentre il tempo impiegato per la creazione del grafico è di circa 45 secondi, per un totale di 205 secondi circa.
La stima dell'incertezza dei sinonimi è stata limitata proponendo all'utente 5 sinonimi differenti,ovvero 5 parole con lo score di similarità maggiore rispetto alla parola data.

Valutazione
La valutazione è stata effettuata prendendo in considerazione un insieme di test composto da 20 frasi. Per ogni frase, è stato assegnato 1 punto per ogni parola per cui è stata predetto almeno un sinonimo corretto ed è stat calcolata l'accuracy. Infine è stata calcolata l'accuracy media tra tutte le frasi ottenendo un valore di 0,65. Per migliorare questo valore sarebbe opportuno effettuare diverse operazioni,come: eliminare i nomi di persona nella fase di pre-processing,inserire una soglia per la similarità di circa 0,5 e ampliare il dataset delle frasi per aumentare il numero di topic ricoperti.

Librerie utilizzate

- *Tkinter* per la creazione dell'interfaccia grafica
- *Pandas* per le operazioni di lettura da file .csv
- *Nltk* per il download del pacchetto di stopwords e le operazioni di text processing
- *Gensim* per la creazione del modello Word2Vec
- *Levenshtein* per la funzione di correzione sintattica
- *Pickle, os* e *time* per utilities
- *Sklearn* e *bokeh* per la creazione del grafico